

REMARKS

Claims 1-9 and 11-71 are pending, of which claims 1, 25, 48, 54, 55, 56, and 61, are independent. In the Office Action, the examiner rejects claims 1-9 and 11-71. Applicants amend claims 1, 25, 48, 54, 55, 56, and 61 herein. No new matter is added. Applicants respectfully request reconsideration of the outstanding rejections and passage of the claims to allowance in light of the following.

Applicants thank the Examiner for withdrawing the finality of the previous Office Action (Office Action at page 2). Further, Applicants thank the Examiner for withdrawing the former 35 U.S.C. §103 rejections (Office Action at page 2).

I. Rejections under 35 U.S.C. §103(a)**A. Claims 1-3, 5-9, 17-22, 24, 25-27, 29-33, 41-46, and 48-71**

In the Office Action, claims 1-3, 5-9, 17-22, 24, 25-27, 29-33, 41-46, and 48-71 were rejected under 35 U.S.C. §103(a) as being unpatentable over MathWorks Simulink® “Dynamic System Simulation for MATLAB,” 1997 (hereinafter “MathWorks”) in view of Official Notice taken, and further in view of U.S. Patent No. 7,107,578 to Alpern (hereafter “Alpern”). Applicants respectfully traverse the rejection.

1. Claims 1-3, 5-9, 17-22, 24, and 70

Applicants’ claim 1 recites,

1. A method comprising the steps of:

*providing a graphical debugger that concurrently interfaces with:
a model view of a model being executed, and
an execution list view of methods called during execution of said model,*

said model comprising a block that includes a plurality of execution methods, said graphical debugger having debug information related to the execution of said model, *said debug information indicating an order of execution of said plurality of execution methods for said block and a start time or a stop time of said plurality of execution methods for said block that are executed during the execution of said model;* and

outputting said debug information to a user, said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of execution methods that are executed in said block during the

execution of said model.

MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest *providing a graphical debugger that concurrently interfaces with a model view of a model being executed and an execution list view of methods called during an execution of the model*, nor do the cited references disclose or suggest *said debug information indicating order of execution of said plurality of execution methods for said block and a start time or a stop time of said plurality of execution methods for said block that are executed during the execution of said model*, as present in claim 1.

In order to better clarify claim 1, Applicants amend claim 1 to recite *providing a graphical debugger that concurrently interfaces with a model view of a model being executed and an execution list view of methods called during an execution of the model*. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 1.

Alpern is generally directed to debugging a computer program that is executed by multiple virtual machines (Alpern at col. 1, lines 15-20). Alpern does not debug *a model being executed*, but rather debugs modules of code written in multiple programming languages, each module running on separate virtual machines (Alpern at col. 11, lines 17-20). In contrast, claim 1 recites a debugger that is *concurrently interfaced with a model view of a model being executed and an execution list view of methods called during an execution of the model*.

There is no indication that the debugger of Alpern could be applied in a block diagram modeling situation, as described in claim 1. The present Specification describes the limitations of current debuggers at pages 3-4. For example, at page 4, the Specification notes that a block diagram as designed by a user may be modified as part of compilation before it is simulated. The compilation process may add or remove elements to optimize simulation. Thus, a conventional debugger applied to a block diagram model may not capture the execution methods for a block being executed, but rather may deal with “simulation information embodied by the compiled version of the block diagram.” The method of claim 1 allows a user to peer into a block in a model being executed to view the block’s execution methods. As described in the Specification at page 5, the debugger of the present Application may provide such a capability

by, for example, interfacing with both a *model* and an *execution list*. There is no indication that a debugger such as Alpern's could be made to interface with a model, such as a block diagram model. Alpern does not disclose or suggest ***providing a graphical debugger that concurrently interfaces with a model view of a model being executed and an execution list view of methods called during an execution of the model***, as recited in amended independent claim 1.

Claim 1 recites still other features, such as ***said debug information indicating order of execution of said plurality of execution methods for said block and a start time or a stop time of said plurality of execution methods for said block that are executed during the execution of said model***. The Examiner correctly notes that the combination of MathWorks and the Official Notice does not disclose ***said debug information indicating order of execution of said plurality of execution methods for said block and a start time or a stop time of said plurality of execution methods for said block that are executed during the execution of said model*** (Office Action at page 3). The Examiner then asserts that Alpern discloses this feature of claim 1 (Office Action at page 4). Applicants respectfully disagree with the Examiner's assertion regarding this feature of claim 1. Instead, the Examiner relies on Alpern for such a disclosure. However, Alpern does not disclose or suggest this feature of claim 1.

The Examiner suggests that Alpern discloses the above feature of claim 1 at Figure 5 and the accompanying description, and col. 12, lines 35-46 (Office Action at page 4). The Examiner quotes Alpern at col. 12, lines 35-46, which states that Alpern "shows the sequence of frames representing routines in the order executed." Figure 5 depicts a stack which includes frames from modules executed by a virtual machine (Alpern at col. 12, lines 26 – 29).

However, Alpern does not disclose ***debug information indicating order of execution of said plurality of execution methods for said block and a start time or a stop time of said plurality of execution methods for said block that are executed during the execution of said model***. Neither the frames, nor the routines, of Alpern are execution methods ***for a block***. Rather, the frames and routines represent "modules of coded instructions for different virtual machines based on modules in different high-level languages" (Alpern at col. 11, lines 17-20). For example, in Figure 5, the stack includes JAVA frames and SQL frames (Alpern at col. 12, lines 30-32). These frames represent, for example, SQL statements executed by the SQL virtual machine of a database server, where the program being debugged involves modules written in SQL as well as JAVA (Alpern at col. 12, lines 30-34 and col. 11, lines 14-23). In contrast, as

recited in claim 1, the **block** that the execution methods are for is part of ***a model being executed, said model comprising a block that includes a plurality of execution methods***. The “routines in the order executed” of Alpern do not constitute ***execution methods for said block***, because Alpern is not concerned with a “block” in a “model.”

For at least the reasons given above, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of independent claim 1. Claims 2-3, 5-9, 17-22, 24, and 70 depend from claim 1 and, as such, include each and every element of claim 1. Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of claims 2-3, 5-9, 17-22, 24, and 70. Therefore, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 1-3, 5-9, 17-22, 24, and 70 be withdrawn.

2. Claims 25-27, 29-33, 41-46, and 71

Applicants’ claim 25 recites,

25. A medium for use in a modeling and execution environment on an electronic device, said medium holding executable instructions on the electronic device for performing an execution method, said method comprising the steps of:

***providing a graphical debugger that concurrently interfaces with:
a model view of a model being executed, and
an execution list view of methods called during an execution of the
model,***

said model comprising a block that includes a plurality of execution methods, said graphical debugger having debug information related to the execution of said model, ***said debug information indicating at least one of the order of the execution of a plurality of execution methods in said model and a start time or a stop time of at least one execution method executed during the execution of said model;*** and

outputting said debug information to a user, said debug information allowing the user to determine proper or improper operation for at least a subset of said plurality of execution methods for the block that are executed during the execution of said model.

In order to further clarify claim 25, Applicants amend claim 25 to recite ***providing a graphical debugger that concurrently interfaces with a model view of a model being executed and an execution list view of methods called during an execution of the model***. Applicants

respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above feature of amended independent claim 25.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest *said debug information indicating at least one of the order of the execution of a plurality of execution methods in said model and a start time or a stop time of at least one execution method executed during the execution of said model*, as present in claim 25.

Neither MathWorks nor the Official Notice disclose or suggest the above feature of claim 25. Specifically, MathWorks and the Official Notice do not disclose or suggest *information indicating at least one of the order of the execution of a plurality of execution methods in said model*. MathWorks and the Official Notice are not concerned with debugging at the level of execution methods in the model.

The addition of the Alpern does not cure the factual defects of MathWorks with respect to claim 25. As noted above in relation to claim 1, Alpern is not concerned with blocks in a model. There is no indication that Alpern could be modified to interface with a model in order to *indicate at least one of the order of the execution of a plurality of execution methods in said model and a start time or a stop time of at least one execution method executed during the execution of said model*.

Claims 26-27, 29-33, 41-46, and 71 depend from claim 25, and thus include each and every element of claim 25. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 25-27, 29-33, 41-46, and 71 be withdrawn.

3. Claims 48-53

Applicants' claim 48 recites,

48. A computer-implemented method, comprising:
- identifying a first execution method operating in a first environment of a computer-based modeling application that executes a model, where the first environment is one of a text-based environment, a time-based block diagram, a state based block diagram, or a data-flow diagram;
 - identifying a second execution method operating in a second environment,

where the second environment differs from the first environment;

debugging the first execution method and the second execution method with a debugger that is concurrently interfaced with the first environment and the second environment while the computer-based model operates on behalf of a user; and

generating output information for the user or for a destination, the output information identifying when the first execution method or the second execution method are operating, identifying an operation performed by the first execution method or the second execution method at a determined location in the first execution method or the second execution method, or identifying an error related to the first execution method or the second execution method during execution of the computer-based model.

In order to further clarify claim 48, Applicants amend claim 48 to recite *debugging the first execution method and the second execution method with a debugger that is concurrently interfaced with the first environment and the second environment while the computer-based model operates on behalf of a user*. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 48.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest *debugging the first execution method and the second execution method with a debugger that is simultaneously interfaced with the first environment and the second environment while the computer-based model operates on behalf of a user*, or *generating output information for the user or for a destination, the output information identifying when the first execution method or the second execution method are operating, identifying an operation performed by the first execution method or the second execution method at a determined location in the first execution method or the second execution method, or identifying an error related to the first execution method or the second execution method during execution of the computer-based model*, as present in amended independent claim 48.

As noted earlier in the claim, the *first execution method operates in a first environment of a computer-based modeling application that executes a model*. As previously noted, MathWorks and the Official Notice taken are not concerned with debugging at the level of an execution method in a modeling application.

Further, Alpern does not allow an execution method that *operates in a first environment of a computer-based modeling application that executes a model* to be debugged. Alpern is concerned with debugging modules for virtual machines that were written in different languages. There is no indication that the methods Alpern utilizes would allow Alpern's debugging method to be applied in *a first environment of a computer-based modeling application that executes a model*. As noted above in relation to claim 1, there are a number of difficulties with debugging in such an environment which Alpern does not address. Because Alpern does not address debugging an execution method in a first environment of a computer-based modeling application that executes a model, Alpern does not disclose or suggest *generating output information for the user or for a destination, the output information identifying when the first execution method or the second execution method are operating, identifying an operation performed by the first execution method or the second execution method at a determined location in the first execution method or the second execution method, or identifying an error related to the first execution method or the second execution method during execution of the computer-based model*, as present in claim 48.

Therefore, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of independent claim 48. Claims 49-53 depend from claim 48, and thus include each and every element of claim 48. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 48-53 be withdrawn.

4. Claim 54

Claim 54 recites,

54. A method, comprising:
receiving information about a first execution method and a second execution method on behalf of a graphical model comprising blocks, where at least one of the blocks includes the first execution method and at least one other execution method or the second execution method and the at least one other execution method, where the first execution method or the second execution method are related to one or more of the blocks;
identifying at least a portion of the first execution method or the second execution method when the first execution method or the second execution method are running, respectively;

obtaining information about the running of the first execution method or the second execution method using the identifying; and
providing debugging information to a user via a display or providing debugging information to a destination device, ***the debugging information provided by a debugger that concurrently interfaces with the graphical model and a list of execution methods called during simulation of the graphical model***, the debugging information identifying the first execution method or the second execution method and information about the first execution method or the second execution method, respectively

The Examiner states, “as per claim 54, note the rejection of claim 1 above. The Instant Claim recites substantially same (sic) limitations as the above-rejected claims and therefore rejected (sic) under same prior-art teachings” (Office Action at page 8). However, claims 1 and 54 recite very different limitations. The Examiner has not considered claim 54 and improperly rejects the claim.

In order to further clarify the claim, Applicants amend claim 54 to recite ***the debugging information provided by a debugger that concurrently interfaces with the graphical model and a list of execution methods called during simulation of the graphical model***. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 54.

Further, even if the rejection of claim 54 were proper, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest at least ***receiving information about a first execution method and a second execution method on behalf of a graphical model comprising blocks, where at least one of the blocks includes the first execution method and at least one other execution method or the second execution method and the at least one other execution method***, as present in claim 54. MathWorks and the Official Notice taken are not concerned with execution methods included in blocks. Further, as noted above, Alpern does not address a ***graphical model comprising blocks***, and there is no indication that Alpern could be modified to work with such a graphical model.

Therefore, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose each and every element of claim 54. Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claim 54 be withdrawn.

5. **Claim 55**

Claim 55 recites,

55. A method, comprising:
identifying a first root method comprising one or more child methods, the first root method related to a graphical modeling application;
identifying a second root method related to the graphical modeling application;
running the first root method and the second root method in a graphical debugger to obtain information about the operation of the first root method or the second root method, *the graphical debugger concurrently interfaced with a method list of methods called during simulation of a model in the graphical modeling application and a model view of a model in the graphical modeling application*; and
displaying a debugging result to a destination, *the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method.*

In order to further clarify claim 55, Applicants amend claim 55 to recite that *the graphical debugger is concurrently interfaced with a method list of methods called during simulation of a model in the graphical modeling application and a model view of a model in the graphical modeling application*. Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 55.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest *the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method*, as present in claim 55.

MathWorks does not address debugging root methods, because MathWorks is focused at

the block level and not the level of root methods. Thus, MathWorks does not give a debugging result with visual identifiers related to the operation of root or child methods (see, e.g., MathWorks at page 12-16). MathWorks does not disclose or suggest displaying status information related to root or child methods, but can display status information related to blocks and overall system states (MathWorks at pages 12-12, 12-14, and 12-16). The Official Notice taken is also not concerned with root methods, and so does not disclose or suggest the above-cited feature of claim 55.

Alpern also does not disclose or suggest *the debugging result comprising visual identifiers related to the operation of the first root method, the one or more child methods or the second root method, error information about the first root method, the one or more child methods or the second root method, an execution result for the first root method, the one or more child methods or the second root method, or status information related to the first root method, the one or more child methods or the second root method*. As noted in a preceding clause in claim 55, the first and second root methods are *related to a graphical modeling application*. Alpern does not suggest a way for a debugging method to be applied to a graphical modeling application.

Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest all the elements of claim 55. Therefore, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claim 55 be withdrawn.

6. Claims 56-60

Claim 56 recites,

56. A method for implementing a user interface for debugging a graphical model, the method comprising:

displaying a hierarchy comprising information about a first root method, one or more child methods related to the first root method, or a second root method, the hierarchy displaying information about the first root method, the one or more child methods, or the second root method in an arrangement representing a relationship among the first root method, the one or more child methods, or the second root method; and

displaying an indicator on the hierarchy proximate to the first root method, the one or more child methods, or the second root method, the indicator denoting a status of the first root method, the one or more child methods, or the second root method, where the status indicates whether the first root method, the one or more

child methods, or the second root method are operating according to determined parameters, ***the determination of whether the first root method, the one or more child methods, or the second root method are operating according to determined parameters made by a debugger that concurrently interfaces with a method list of methods called during simulation of the graphical model a model view of the graphical model.***

In order to further clarify claim 56, Applicants amend claim 56 to recite that ***the determination of whether the first root method, the one or more child methods, or the second root method are operating according to determined parameters is made by a debugger that concurrently interfaces with a method list of methods called during simulation of the graphical model a model view of the graphical model.*** Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 56.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest ***displaying a hierarchy comprising information about a first root method, one or more child methods related to the first root method, or a second root method, the hierarchy displaying information about the first root method, the one or more child methods, or the second root method in an arrangement representing a relationship among the first root method, the one or more child methods, or the second root method,*** as present in claim 56.

MathWorks and the Official Notice taken are concerned with block-level debugging. Instead of displaying a hierarchy of root and child methods, MathWorks displays a block execution order, and not information relating to root and child methods. For instance, compare MathWorks at page 12-16, showing the hierarchy of MathWorks, with Figure 18A of the Application, which depicts the parent-child hierarchy. The block execution order of MathWorks is silent as to root or child methods.

Moreover, Alpern does not address root and child methods as described in claim 56. In fact, the frames and execution methods that the Examiner cites in Alpern come from *different modules programmed in different languages* (Alpern at col. 11, lines 14-23). Because they are programmed separately and in different languages, there is no root or child method relationship between the different frames in Figure 5 of Alpern.

Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest all the elements of claim 56. Claims 57-60 depend from claim 56, and thus include every element of claim 56. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 56-60 be withdrawn.

7. **Claims 61-69**

Claim 61 recites,

61. A method for debugging operation of a graphical icon, the method comprising:
identifying a plurality of execution methods for the graphical icon using a plurality of regions related to the graphical icon;
displaying information about a first one of the plurality of execution methods in a first one of the plurality of regions or information about a second one of the plurality of execution methods in a second one of the plurality of regions; and
associating the information in the first one of the plurality of regions or information in the second one of the plurality of regions with a graphical debugger to provide a user with debugging results for the first one of the plurality of execution methods or the second one of the plurality of execution methods, the debugging results allowing the user to identify desirable operations performed on behalf of the graphical icon or undesirable operations performed on behalf of the graphical icon, *the graphical debugger concurrently interfacing with a view of the graphical icon and an execution methods list of execution methods performed on behalf of the graphical icon.*

In order to further clarify claim 61, Applicants amend claim 61 to recite that *the graphical debugger concurrently interfaces with a view of the graphical icon and an execution methods list of execution methods performed on behalf of the graphical icon.* Applicants respectfully submit that MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest the above-quoted feature of amended independent claim 61.

Further, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest *identifying a plurality of execution methods for the graphical icon using a plurality of regions related to the graphical icon*, nor does MathWorks disclose or suggest *displaying information about a first one of the plurality of execution*

methods in a first one of the plurality of regions or information about a second one of the plurality of execution methods in a second one of the plurality of regions, as present in claim 61.

MathWorks does not display information about execution methods. MathWorks obtains information only on individual block states and overall system states (MathWorks at pages 12-12 to 12-14). Thus, MathWorks does not identify execution methods for a graphical icon, as present in claim 61. MathWorks is also silent as to displaying information about execution methods in a plurality of regions, as present in claim 61. The Official Notice taken is also not concerned with debugging execution methods.

Alpern also does not *identify a plurality of execution methods for the graphical icon using a plurality of regions related to the graphical icon*. Alpern debugs modules of code written in different languages, and is not concerned with graphical icons. For example, in Figure 5, Alpern depicts JAVA and SQL snippets, and not graphical icons.

Thus, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest all the elements of claim 61. Claims 62-69 depend from claim 61, and thus include every element of claim 61. In light of the above arguments, Applicants respectfully request that the 35 U.S.C. §103(a) rejection of claims 61-69 be withdrawn.

B. Claims 4, 10-16, 23, 28, 34-40, and 47

Claims 4, 10-16, 23, 28, 34-40, and 47 were rejected under 35 U.S.C. §103(a) as being unpatentable over MathWorks in view of the Official Notice taken and Alpern, and further in view of Fenlason's "GNU gprof" (1998). Applicants respectfully traverse the rejection.

Claims 4, 10-16, and 23 depend from claim 1 and, as such, include each and every element of claim 1. Claims 28, 34-40, and 47 depend from claim 25 and, as such, include each and every element of claim 25. As noted above, MathWorks, the Official Notice taken, and Alpern, alone or in any reasonable combination, do not disclose or suggest each and every element of claims 1 and 25.

The addition of GNU gprof does not cure the factual deficiencies of MathWorks, the Official Notice taken, and Alpern in regards to claims 1 and 25. GNU gprof is a profiler used to

determine which parts of a program are taking the most execution time (GNU gprof at 1). GNU gprof is used to profile programs, not models including blocks. Thus, GNU gprof does not disclose or suggest *a debugger interfaced with a model view of a model being executed*, said model comprising *a block*, nor does GNU gprof provide debug information indicating the order of execution of execution methods *for blocks*. GNU gprof does not disclose execution methods in blocks, a feature also missing from MathWorks, the Official Notice taken, and Alpern. GNU gprof also does not disclose a debugger that is capable of simultaneously interface with a model view of a model being executed and an execution list. Therefore, GNU gprof does not remedy the shortcomings of MathWorks with respect to at least the above-mentioned features of claims 1 and 25.

For at least the reasons presented above, MathWorks, the Official Notice taken, Alpern, and GNU gprof, alone or in any reasonable combination, do not disclose each and every element of independent claim 1. Claims 4, 10-16, and 23 depend from claim 1 and, as such, include each and every element of claim 1. Claims 28, 34-40, and 47 depend from claim 25 and, as such, include each and every element of claim 25. Therefore, MathWorks, the Official Notice taken, Alpern, and GNU gprof do not disclose each and every element of claims 4, 10-16, 23, 28, 34-40, and 47. Applicants therefore respectfully request that the 35 U.S.C. §103(a) rejection of claims 4, 10-16, 23, 28, 34-40, and 47 be withdrawn.

CONCLUSION

Applicants believe the pending application is in condition for allowance. If the Examiner feels that further discussion would expedite the proceedings, the Examiner is urged to call Applicants' attorney at the phone number listed below.

Please charge any shortage or credit any overpayment of fees to our Deposit Account No. 12-0080, under Order No. MWS-106RCE. In the event that a petition for an extension of time is required to be submitted herewith, and the requisite petition does not accompany this response, the undersigned hereby petitions under 37 C.F.R. §1.136(a) for an extension of time for as many months as are required to render this submission timely. Any fee due is authorized to be charged to the aforementioned Deposit Account.

Dated: December 23, 2008

Respectfully submitted,

Electronic signature: /Kevin J. Canning/
Kevin J. Canning
Registration No.: 35,470
LAHIVE & COCKFIELD, LLP
One Post Office Square
Boston, Massachusetts 02109-2127
(617) 227-7400
(617) 742-4214 (Fax)
Attorney/Agent For Applicant